Technical Alert 21-002: Impact of Changes of Synchronous XMLHttpRequest in Google Chrome and Microsoft Edge on IMI Courseware

04 October 2021

The Army Distributed Learning Program, Army University
Bldg. 2112, Pershing Ave.
Fort Eustis, VA 23604-1412

Table of Contents

Introduction

This Technical Alert provides information about Synchronous XMLHttpRequest (XHR)-related browser changes and their impact on existing and future courseware. Google Chrome and Microsoft Edge now disallow synchronous XHR requests upon page dismissal. As a result, student learning progress data frequently fails to save properly on Learning Management Systems when courseware runs in these browsers. Courseware is not affected when running in Internet Explorer. The TA will also provide recommendations for resolving related courseware/LMS communication problems.

Browser Setting Changes

To mitigate the risk of losing learner progress data when courseware runs in Google Chrome or Microsoft Edge, users can enable the flag “Allows synchronous XHR requests in page dismissal” in the Chrome or Edge browser using the following steps:

1. In the address bar, type “`about://flags/#allow-sync-xhr-in-page-dismissal`” (or type “`about:flags`” and search “Allows synchronous XHR requests in page dismissal”)
2. Select “Enabled” beside “Allows synchronous XHR requests in page dismissal”
3. Restart the browser

Note: The enabled flag discussed above applies to all sites to be visited in the browser. This is a temporary measure, which is expected to be removed in Chrome version 88.

Instructions for User/Learner

Loss of learner progress data can also occur when the learner exits the content using the “x” of the browser window instead of an “Exit” button programmed within the courseware. All courseware accessed on the ALMS must contain an “Exit” button on each page of content to mitigate this risk. This is a formal requirement in the 2021 BRBP. The Letter of Instruction (LOI) should advise learners to click only the “Exit” button in the courseware instead of clicking the” x” button on the top right corner of the learning content window to exit the content.

The above instructions for the learner to exit the learning content by clicking the “Exit” button may be repeated in on-screen messages so that learners are reminded multiple times to exit the course properly.

Courseware Changes

Business Rule 2.2.1 in the Army DL BRBP 2021 requires a two-window structure for courses deployed on the ALMS. In the two-window structure, the parent window (ALMS SCORM player window) is often used only to launch the child/content window. However, the parent window is a safer place to handle SCORM communication. The parent window resides in the SCORM player and launches the learning content in a child window. The parent window displays a message informing the learner to keep the SCORM player window open while the child window contains the actual learning content. Since the parent window has the warning message, it is less likely for the user to force the window closure by clicking the window’s built-in “x” button. In addition, the parent window is a good place for important instructions, such as only using the “Exit" button to exit the course.

If an authoring tool is used, courseware developers may choose to patch the template code included in the authoring tool so that the parent SCORM player window handles SCORM communication while the child window presents the learning content.

Conclusions

This document presents the potential learning data loss triggered by the changes in Chrome and Edge browsers and solutions to mitigate the learning data loss risks. Courseware developers may choose the solutions most suitable for their situations.

Supplemental Technical Information for DL Developers Mechanism of Submitting Learning Data

SCORM 2004 Run-Time Environment (RTE) Application Programming Interface (API) provides two methods for SCOs to send data to LMS:

- **`SetValue`**: allows the SCO to send information to the LMS for storage. The API Instance may be designed to immediately persist data that was set (to the server-side component) or store data in a local (client-side) cache.
- **`Commit`**: requests forwarding to the persistent data store any data from the SCO that may have been cached by the API Instance since the last call to `Initialize(“”)` or `Commit(“”)`, whichever occurred most recently. The LMS would then set the error code to `0` (No Error encountered) and return `“true”`. If the API Instance does not cache values, `Commit(“”)` shall return “true” and set the error code to `0` (No Error encountered) and do no other processing.

There are two options for the implementation when the SCO actually sends the data: (1) sends at each `SetValue` call, or (2) sends at the next `Commit` call with the cached, accumulated data set by (possibly multiple) `SetValue` calls. The ALMS adopts the second option for the sake of efficiency.

The definition indicates that both `SetValue` and `Commit` are **blocking** or **synchronous** methods: the program must wait until it has received the server’s response to the data submission requests. Therefore, most of the current LMS implementations including ALMS use **synchronous** `XMLHttpRequest` (XHR) to send the requests. This implementation option will become obsolete because future web platforms will eventually deprecate the use of synchronous XHR on the main JavaScript thread due to its detrimental effects to the end user’s experience: users are unable to interact with the pages during data transmission. Major browsers, including Chrome, Edge, and Firefox, have deprecated (although not prohibited) synchronous requests on the main thread. Starting with version 80, Chrome disallows it during page dismissal when navigating away from or closing the page by the user. This applies to `beforeunload`, `unload`, `pagehide`, and `visibilitychange` events. Microsoft Edge behaves similarly since it based on Chromium, the same core of Chrome.